Stackelberg solutions in macroeconometric policy models with a decentralized decision structure I

de Zeeuw, A.J.

Publication date:
1979

Citation for published version (APA):
de Zeeuw, A. J. (1979). *Stackelberg solutions in macroeconometric policy models with a decentralized decision structure I*. (Research memorandum / Tilburg University, Department of Economics; Vol. FEW 84). Unknown Publisher.

subfaculteit der econometrie

RESEARCH MEMORANDUM

[redacted]

TILBURG UNIVERSITY
DEPARTMENT OF ECONOMICS
Postbus 90135 - 5000 LE Tilburg
Netherlands

[redacted]

[redacted]

FEW

84

Stackelberg Solutions in Macroeconometric Policy Models with a Decentralized Decision Structure I.

Aart J. de Zeeuw

June 1979

Tilburg University, Department of Econometrics.

R 11

T economic policy
T game theory

This research was supported by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Abstract

Macroeconometric policy models with a decentralized decision structure can often be viewed upon as a linear quadratic N-person nonzero sum difference game with exogenous inputs, nonfeasible ideal paths and a fixed time horizon. When the decision structure is also hierarchical, the Stackelberg solution concept can be used to model the decision structure. In this paper the open loop and feedback Stackelberg strategies for such a game are derived.

Contents

Key Words

L. Q.-difference games, Stackelberg solutions, linked macroeconometric policy models.

1. Introduction

Macroeconometric policy models with a decentralized decision structure can often be converted into a linear quadratic nonzero sum difference game with exogenous inputs, nonfeasible ideal paths and a fixed time horizon (see [1]).
The formal framework for this game in terms of general systems theory can be found in [2].
When it is assumed that the players have a competitive mood of play and expect rational behaviour of the other players, the Nash solution concept can be used to model the decision structure ([1]).
When it is assumed that the players have a cooperative mood of play, Pareto is the appropriate solution concept ([2]).
When some players are dominated by others, either due to a lack of information about the performance functionals of other players or due to differences in size or strength, the Stackelberg solution concept can be used. This concept presupposes a leader-follower decision structure and rational behaviour of the followers.
The main ideas for this paper stem from [3] and [4].
The references [6], [7] and [8] can also be clarifying.
A survey of results for the Stackelberg solution concept can be found in [5].

2. Definition and Example.

When the players in a game announce their strategies one after another, we call their optimal strategies the Stackelberg solution for the game. This solution concept comes to mind, when we want to model the decision structure in a game as being hierarchical or sequential, having a leader-follower structure in some sense. It is assumed, that every player expects his followers to behave rationally.

Suppose we have an N-player game, where player N is the leader and where players N-1 to 1 follow one after another. Let U_1 to U_N be the sets

of admissable strategies for player 1 to N, respectively, and let
$J_i : U_1 \times U_2 \times \ldots \times U_N \to \mathbb{R}^+$, $i = 1,2, \ldots, N$,
be the cost functionals, that players 1 to N, respectively, want to minimize. Stackelberg solutions for this game can be defined as follows.

Definition:

If there exist mappings $T_i : U_{i+1} \times U_{i+2} \times \ldots \times U_N \to U_i$, $i = 1,2, \ldots, N-1$, such that for any fixed $(u_{i+1}, u_{i+2}, \ldots, u_N) \in U_{i+1} \times U_{i+2} \times \ldots \times U_N$

i) $(u_1, u_2, \ldots, u_{i-1}, T_i((u_{i+1}, u_{i+2}, \ldots, u_N)), u_{i+1}, u_{i+2}, \ldots, u_N) \in D_{i-1}$

ii) $J_i((u_1, u_2, \ldots, u_{i-1}, T_i((u_{i+1}, u_{i+2}, \ldots, u_N)), u_{i+1}, u_{i+2}, \ldots, u_N)) \leq$

$\leq J_i((u_1, u_2, \ldots, u_N))$ for all $(u_1, u_2, \ldots, u_N) \in D_{i-1}$,

where $D_o = U_1 x U_2 x \ldots x\ U_N$,

$D_i = \{(u_1, u_2, \ldots, u_N) \in D_{i-1} \mid u_i = T_i((u_{i+1}, u_{i+2}, \ldots, u_N))\}$,

$i = 1,2,\ldots,N-1$,

and if there exists a $(\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_N) \in D_{N-1}$, such that

$J_N((\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_N)) \leq J_N((u_1, u_2, \ldots, u_N))$ for all $(u_1, u_2, \ldots, u_N) \in D_{N-1}$

then $(\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_N)$ is called a Stackelberg solution for the game.

In the case of two players we can immediately conclude from this definition and from the definition of a Nash solution (see e.g. [1]), that, because Nash solutions belong to D_1, Stackelberg solutions are favourable to the leader as compared to Nash solutions (see also [6]).

To find a Stackelberg solution for the game we proceed, according to the definition, as follows. At first we express the rational behaviour of player 1. Given this rational behaviour, we look for the rational behaviour of player 2 and so on until we can solve for the optimal strategy of player N, the leader.
To illustrate the concept of a Stackelberg solution we use the famous two person nonzero sum static game, represented by figure (a).

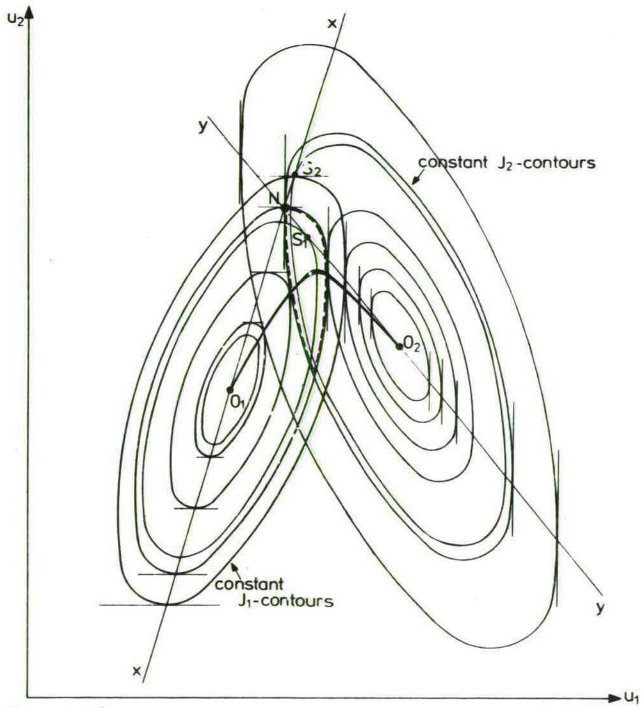

Figure a)

On the axes we find the admissable strategies for player 1 and 2, respectively.
The curves are isocost contours.
The lines xx and yy are rational behaviour lines for player 1 and 2, respectively.
The Nash solution N is the intersection of the lines xx and yy.

The Pareto set of noninferior solutions (see e.g. [2]) is given by the dark line connecting 0_1 and 0_2.
The shaded area represents the set of solutions that are favourable to both players as compared to the Nash solution.
The Stackelberg solution S_2 for the game with player 2 as leader is the tangent point of the line xx with the set of isocost contours for player 2. Similarly, the Stackelberg solution S_1 for the game with player 1 as leader is the tangent point of the line yy with the set of isocost contours for player 1.
Note that both S_1 and S_2 are on a lower cost contour than N, when we consider the leader of the game. Interesting is, that S_1 lies in the shaded area, which means that the follower has also lower costs than in N, whereas S_2 lies outside the shaded area, which means that the follower is worse off than in N.
Other examples can be found in [6], [7] and [8].

3. The Model.

We consider a macroeconometric policy model with N policy makers. The objectives of these policy makers are assumed to be in the form of N quadratic cost functionals $J_1, J_2, .., J_N$, that weigh deviations of objective variables and controlvariables from ideal paths, set by the different policy makers, and sum those costs over a planning period.
All the policy makers have the same view of reality; that is all the cost functionals are constrained by the same model. This model is converted into a system in state space form. For mathematical reasons the system equations have to be linearized in the state variables and the controlvariables. Econometric models are mostly specified in discrete time, so the time parameter is discrete. Some exogenous inputscenarios influence the system. The ideal paths generally do not form a solution for the system equations.
The N policy makers are not equally powerful: we assume a decision hierarchy. The strategies are chosen one after another. To be able to select an optimal strategy, each player must have an expectation of the behaviour of his followers: rational behaviour is assumed.

A model like this is called a linear quadratic nonzero sum difference game with exogenous inputs, nonfeasible ideal paths and a fixed time horizon, subject to the Stackelberg solution concept.

Formally:

$$J_i(x_o,u_1(.),u_2(.),..,u_N(.)) =$$

$$= \sum_{t=t_o}^{t_f} \tfrac{1}{2}[y(t)-\hat{y}_i(t)]'Q_i(t)[y(t)-\hat{y}_i(t)] + \sum_{t=t_o}^{t_f-1} \sum_{j=1}^{N} \tfrac{1}{2}[u_j(t)-\hat{u}_j(t)]'R_{ij}(t)[u_j(t)-\hat{u}_j(t)]$$

$i= 1,2,..,N,$

subject to the system

$$\begin{cases} x(t+1)-x(t)=A(t)x(t)+\sum_{i=1}^{N} B_i(t)u_i(t)+C(t)z_1(t), t=t_o,t_o+1,..,t_f-1, \\ x(t_o)=x_o \\ y(t)=F(t)x(t)+G(t)z_2(t), t=t_o,t_o+1,..,t_f, \end{cases}$$

where

$x(.)$ are the state variables,

$u_i(.)$ are the controlvariables,

$y(.)$ are the objective variables,

$z_1(.)$ and $z_2(.)$ are the exogenous variables,

$\hat{y}_i(.)$ and $\hat{u}_i(.)$ are the ideal paths for the objective variables and controlvariables, respectively, set by player i,

$Q_i(.)$: (mxm) matrix,

$R_{ij}(.)$: $(s_i x s_i)$ matrix,

$A(.)$: $(n x n)$ matrix,

$B_i(.)$: $(n x s_i)$ matrix,

$C(.)$: $(n x r_1)$ matrix,

$F(.)$: $(m x n)$ matrix,

$G(.)$: $(m x r_2)$ matrix.

It is natural to assume the matrices $Q_i(.)$ and $R_{ij}(.)$ to be positive semi-definite. To avoid singularities we assume the matrices $R_{ii}(.)$ to be positive definite.
Without loss of generality we assume the matrices $Q_i(.)$ and $R_{ij}(.)$ to be symmetric.
Player N selects his strategy first, then player N-1, and so on.

4. The Principle of Optimality.

It is important to check , whether the Stackelberg solution concept satisfies Bellman's principle of optimality, because, if it is satisfied, we can apply the dynamic programming technique to find a (closed loop no memory) solution for the game.
The principle of optimality states, that an optimal strategy for the problem starting in x_o at t_o must have the property, that the part of the strategy, that remains after having applied it some time steps, is also optimal for the problem starting in the state reached at that point in time.
The Stackelberg solution does not satisfy the principle of optimality. We will show, why this is the case for the model described above, and we will give a numerical example.
The rational behaviour of player 1 can be found by solving a standard optimal control problem, because we can consider the strategies of the other players as exogenous inputs. This leads to (see e.g. [2]) :

$$u_1(t)=-(R_{11}(t)+B_1'(t)K_1(t+1)B_1(t))^{-1}B_1'(t)(K_1(t+1)((I+A)x(t)+$$

$$+\sum_{i=2}^{N} B_i(t)u_i(t)+C(t)z_1(t))+g_1(t+1))+\hat{u}_1(t),t=t_o,t_o+1,\dots,t_f-1$$

where

$$g_1(t)=L_1(g_1(t+1),u_2(t),u_3(t),\dots,u_N(t)),t=t_o+1,t_o+2,\dots,\ t_f-1$$

It follows, that

$$g_1(t)\ =L_2(u_2(t),u_2(t+1),\dots,u_2(t_f-1),u_3(t),\dots,u_N(t_f-1)),t=t_0+1,t_o+2,\dots,t_f-1$$

When we substitute the rational behaviour of player 1 into the system equations, we find

$$x(t+1)=L_3(x(t),u_2(t),u_2(t+1)\dots,\ u_2(t_f-1),u_3(t),\dots,u_N(t_f-1)),$$

$$t=t_o,t_o+1,\dots,t_f-1.$$

Observe, that we don't have a system anymore, because the axiom of determinism or the non-anticipativity is no longer satisfied. It is clear, that in a situation like this the principle of optimality can't be satisfied.

Example:

$N=2,\ T=\{t_o,t_1,t_2\};A(.)=0;C(.)=0;F(.)=1;$

$B_1(t_o)=1,B_1(t_1)=0,B_2(t_0)=0,B_2(t_1)=1;$

$Q_1(t_2)=2;Q_2(t_1)=2;R_{11}(t_o)=2;R_{22}(t_1)=2$

The other cost matrices and the ideal paths are zero.

Or: $J_1=x^2(t_2)+u_1^2(t_o)$

$J_2=x^2(t_1)+u_2^2(t_1)$

$x(t_o)=x_o$

$x(t_1)=x(t_o)+u_1(t_o)$

$x(t_2)=x(t_1)+u_2(t_1)$

The rational behaviour of player 1 can be expressed as

$u_1(t_o)=-\frac{1}{2}(x_o+u_2(t_1))$

So, player 2 faces the problem:
minimize $J_2=x^2(t_1)+u_2^2(t_1)$

subject to $x(t_o)=x_o$

$$x(t_1)=\tfrac{1}{2}x(t_o)-\tfrac{1}{2}u_2(t_1)$$

$$x(t_2)=x(t_1)+u_2(t_1)$$

Open loop solution: $u_2^*(t_1)=\frac{1}{5}\,x_0$.

Linear closed loop no memory solution: $u_2^*(t_1)=\frac{1}{2}x^*(t_1)$.
Both solutions give rise to the same trajectories and the same costs.
So, the open loop Stackelberg solution becomes

$$u_1^*\,(t_o)=-\tfrac{3}{5}x_o,u_2^*(t_1)=\tfrac{1}{5}x_o$$

and the linear closed loop no memory Stackelberg solution becomes

$$u_1^*(t_o)=-\tfrac{3}{5}x_o,u_2^*(t_1)=\tfrac{1}{2}x^*(t_1)$$

with trajectory

$$x^*(t_o)=x_o,x^*(t_1)=\tfrac{2}{5}x_o,x^*(t_2)=\tfrac{3}{5}x_o\ .$$

Whereas, starting in $\frac{2}{5}x_o$ at t_1 (in fact not a game situation anymore), the optimal strategy is $u_2^*(t_1)=0$.
So, the principle of optimality isn't satisfied.
A "tree"-example can be found in [7].

The solution, that is found by means of dynamic programming, is called

the feedback Stackelberg solution.
The interpretation of this solution is, that in this case each player waits with announcing his action until he makes observations about the state of the system, that is realized. In the closed loop no memory solution each player announces at the beginning of the planning period, for each point in time, how he will react to his observations about the state of the system: he selects from all functions of observations and time.
In the feedback solution each player announces at each point in time, how he reacts to his observations about the state of the system, that is realized at that point in time: in fact the principle of optimality is postulated.
In the closed loop no memory solution a player can reach lower costs than in the feedback solution by announcing a strategy, that is on the one hand suboptimal for him for the last part of the game, but on the other hand such that his followers, while minimizing their own costs, will choose strategies, that lead to trajectories, that are so much better for him for the first part of the game, as compared to the feedback solution, that the losses in the last part of the game are more than compensated. But the closed loop no memory solution is not stable in the sense that the followers can't be sure, that the leaders will stick to their announced strategies, when the game gets to the stage, where those strategies are inferior to them. Whether the leaders will keep their (suboptimal) "threats", depends on how they weigh the actual costs by their credibility in the future.
For the feedback Stackelberg solution it isn't true anymore in general, that the leader in the two person game has lower costs than be would have had in the feedback Nash solution: a "tree"-example can be found in [7] .
In the next paragraphs we will present results on the open loop Stackelberg solution, where each player announces at the beginning of the planning period a strategy, which only depends on the initial state of the system and time, and results on the feedback Stackelberg solution for the model, described in paragraph 3. The closed loop Stackelberg solution is still a topic of further research; in [9] it is shown,

that the closed loop no memory Stackelberg strategies in this deterministic setting are not linear.

5. Open Loop Stackelberg Solution.

We start with some remarks on the nature of the problem. When we solve for the rational behaviour of player 1, a standard optimal control problem, and substitute this rational behaviour in the system equations and in the cost functionals for the other players, we end up with cost functionals for the other players, system equations, backward recursive matrix Riccati equations and backward recursive tracking equations, driven by the strategies of the other players (a system!), with a fixed boundary condition at the time horizon.
When we eliminate player 1, using the necessary conditions for his rational behaviour according to Pontryagin's minimum principle, we end up with: cost functionals for the other players, system equations and adjoint system equations, with a boundary condition on state and adjoint state at the time horizon.
Both situations can be viewed upon as the same problem as the original one with one player less, but constrained by a system with a dimension, twice as big as the dimension of the original system, and with mixed boundary conditions. To eliminate player 2 we have to solve this two point boundary value optimal control problem. Again we can choose between techniques, based on the "completion of the square" argument, and Portryagin's minimum principle (see e.g. [2]).
In the same way the other players can successively be eliminated. This leads to a sequence $P_i, i=2,3,..,N$, of two point boundary value optimal control problems.
The open loop Stackelberg solution follows directly from the solution of P_N, the optimal control problem for the leader. The solution of P_N can be found by solving a backward recursive matrix Riccati equation and a backward recursive tracking equation.

The technique we use in this paper is successive application of Pontryagin's minimum principle.

In theorem 1 we state and prove, under the assumption that $(I+A)^{-1}$ exists, how the sequence $P_i, i=1,2,..,N$, is formed. Implicitly one finds the rational behaviour of the followers in terms of adjoint variables. We will use the symbols ρ, y and q for the adjoint variables, that show up as a consequence of the elimination process. In theorem 2 we state and prove, which Riccati equation and tracking equation have to be solved to find a solution for P_N and how the open loop Stackelberg solution relates to the solution of P_N.
Notation: for notational convenience we won't write the time dependency of the matrices.

Theorem 1:
Assuming that $(I+A)$ is nonsingular, the successive elimination of followers in the N-level Stackelberg linear quadratic difference game with an open loop information structure by means of successive application of Pontryagin's minimum principle leads to the following sequence of problems:

P_1: $\underset{u_1(.)}{\text{minimize}}\ J_1$

$$\text{where } J_1 = \sum_{t=t_o}^{t_f} \tfrac{1}{2}[y(t)-\hat{y}_1(t)]'Q_1[y(t)-\hat{y}_1(t)] +$$

$$+ \sum_{t=t_0}^{t_f-1} \sum_{j=1}^{N} \tfrac{1}{2}[u_j(t)-\hat{u}_j(t)]'R_{1j}[u_j(t)-\hat{u}_j(t)],$$

subject to

$$\begin{cases} x(t+1)-x(t)=Ax(t)+\sum_{j=1}^{N} B_j(u_j(t)-\hat{u}_j(t))+\sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t), \\ \qquad\qquad t=t_o,t_o+1,\ldots,t_f-1, \\ x(t_o)=x_o \\ y(t)=F\,x(t) + G\,z_2(t)\ ,\ t=t_o,t_o+1,..,t_f. \end{cases}$$

P_i,=i=2,3,...,N: minimize $\underset{u_i(.)}{J_i}$

<u>where</u> $J_i = \sum_{t=t_o}^{t_f} \frac{1}{2}[y(t)-\hat{y}_i(t)]'Q_i[y(t)-\hat{y}_i(t)] +$

$$+ \sum_{t=t_o}^{t_f-1} \sum_{j=1}^{N} \frac{1}{2}[u_j(t)-\hat{u}_j(t)]'R_{ij}[u_j(t)-\hat{u}_j(t)]$$

<u>subject to</u>
a system, consisting of $(2^{i-2}*n)$ dimensional forward recursive equations and $(2^{i-2}*n)$ dimensional backward recursive equations:

$$(5.1)\begin{cases} \tilde{x}_i(t+1)-\tilde{x}_i(t)=\tilde{A}_i\ \tilde{x}_i(t)+\tilde{S}_i\tilde{p}_i(t+1)+v_i(t),t=t_o,t_o+1,..,t_f-1, \\ \tilde{x}_i(t_o)=[\underset{n}{x_o'}\vdots\underset{((2^{i-2}-1)*n)}{o\ldots o}]' \\ y(t)=[\underset{n}{F}\vdots\underset{((2^{i-2}-1)*n)}{o\ldots o}]\tilde{x}_i(t)+Gz_2(t),t=t_o,t_o+1,\ldots,t_f. \end{cases}$$

and

$$(5.2)\begin{cases} \tilde{p}_i(t)-\tilde{p}_i(t+1) = \tilde{A}_i'\tilde{p}_i(t+1)+\tilde{Q}_i\tilde{x}_i(t)+w_i(t),t=t_f-1,t_f-2,..,t_o, \\ \tilde{p}_i(t_f)=\tilde{Q}_i\tilde{x}_i(t_f)+w_i(t_f) \end{cases}$$

<u>where</u>
(i) $u_{i+1}(.),u_{i+2}(.),..,u_N(.)$ are exogenous to P_i,i=1,2,..,N.

(ii) $u_1(t)=[\underset{n}{-R_{11}^{-1}B_1'}\vdots\underset{((2^{i-2}-1)*n)}{o\ldots o}]\tilde{p}_i(t+1)+\hat{u}_1(t)$,i=2,3,..,N,

$$u_j(t)=[\underset{(2^{j-2}*n)}{o\,..\,o}\vdots\underset{n}{-R_{jj}^{-1}B_j'}\vdots\underset{((2^{i-2}-2^{j-2}-1)*n)}{o\,...\,o}]\,\tilde{p}_i(t+1)+\hat{u}_j(t),\ j=2,3,..,i-1,\ i=3,4,\ldots,N,$$

for $t=t_o,t_o+1,\ldots,t_f-1$

(iii) $\tilde{x}_2(.)\triangleq x(.)$,

$$\tilde{x}_i(.)\triangleq\begin{array}{c}\begin{bmatrix}\tilde{x}_{i-1}(.)\\ y_{i-1}(.)\end{bmatrix}\end{array}\begin{array}{l}(2^{i-3}*n)\\ (2^{i-3}*n)\end{array},\ i=3,4,\ldots,N.$$

(iv) $\tilde{p}_2(.)\triangleq p_1(.)$,

$$\tilde{p}_i(.)\triangleq\begin{bmatrix}\tilde{p}_{i-1}(.)\\ p_{i-1}(.)\\ q_{i-1}(.)\end{bmatrix}\begin{array}{l}(2^{i-3}*n)\\ n\\ ((2^{i-3}-1)*n)\end{array},\ i=3,4,\ldots,N.$$

(v) $v_2(.)\triangleq\sum_{j=2}^{N}B_j(u_j(.)-\hat{u}_j(.))+\sum_{j=1}^{N}B_j\hat{u}_j(.)+Cz_1(.)$,

$$v_i(.)\triangleq\begin{bmatrix}\sum_{j=i}^{N}B_j(u_j(.)-\hat{u}_j(.))+\sum_{j=1}^{N}B_j\hat{u}_j(.)+Cz_1(.)\\ o\\ \vdots\\ o\end{bmatrix}\begin{array}{l}n\\ \\ ((2^{i-2}-1)*n),\ i=3,4,\ldots,N.\\ \end{array}$$

(vi) $w_2(.)\triangleq F'Q_1(Gz_2(.)-\hat{y}_1(.))$,

$$w_i(.)\triangleq\begin{bmatrix}w_{i-1}(.)\\ F'Q_{i-1}(Gz_2(.)-\hat{y}_{i-1}(.))\\ o\\ \vdots\\ o\end{bmatrix}\begin{array}{l}(2^{i-3}*n)\\ n\\ \\ \\ ((2^{i-3}-1)*n)\end{array},\ i=3,4,\ldots,N.$$

(vii)

$$\tilde{A}_i \triangleq \underset{(2^{i-2}*n)}{\mathrm{diag}\,(A, A, \ldots, A)}, \quad i = 2,3,\ldots,N.$$

(viii) $\tilde{S}_2 \triangleq -B_1 R_{11}^{-1} B_1'$,

$$\tilde{S}_i \triangleq \begin{array}{c c} \left[\begin{array}{cc} \tilde{S}_{i-1} & -T_{i-1} \\ & \\ -U_{i-1} & -\tilde{S}_{i-1}' \end{array}\right] & \begin{array}{c} (2^{i-3}*n) \\ \\ (2^{i-3}*n) \end{array} \\ (2^{i-3}*n)(2^{i-3}*n) & \end{array}, \quad i=3,4,\ldots,N,$$

where

$$T_{i-1} \triangleq \begin{array}{cc} \left[\begin{array}{cc} B_{i-1} R_{i-1,i-1}^{-1} B_{i-1}' & 0 \\ 0 & 0 \end{array}\right] & \begin{array}{c} n \\ ((2^{i-3}-1)*n) \end{array} \\ \quad n \qquad\qquad ((2^{i-3}-1)*n) & \end{array}$$

and

U_{i-1} is a direct sum:

$$U_{i-1} \triangleq \mathrm{diag}\,(\underset{n}{B_1 R_{11}^{-1} R_{i-1,1} R_{11}^{-1} B_1'}, \underset{n}{B_2 R_{22}^{-1} R_{i-1,2} R_{22}^{-1} B_2'},$$

$$\underset{n}{B_3 R_{33}^{-1} R_{i-1,3} R_{33}^{-1} B_3'}, \underset{n}{0}, \underset{n}{B_4 R_{44}^{-1} R_{i-1,4} R_{44}^{-1} B_4'},$$

$$\underset{((2^2-1)*n)}{0}, \ldots\ldots\ldots, \underset{n}{B_{i-2} R_{i-2,i-2}^{-1} R_{i-1,i-2} R_{i-2,i-2}^{-1} B_{i-2}'}, \underset{((2^{i-4}-1)*n)}{0}).$$

(ix) $\tilde{Q}_2 \triangleq F'Q_1F$,

$$\tilde{Q}_i \triangleq \begin{array}{cc} \left[\begin{array}{cc} \tilde{Q}_{i-1} & 0 \\ V_{i-1} & -\tilde{Q}'_{i-1} \end{array}\right] & \begin{array}{c} (2^{i-3}*n) \\ (2^{i-3}*n) \end{array} \\ (2^{i-3}*n)(2^{i-3}*n) & \end{array}, \quad i=3,4,..,N,$$

where

$$V_{i-1} \triangleq \begin{array}{cc} \left[\begin{array}{cc} F'Q_{i-1}F & 0 \\ 0 & 0 \end{array}\right] & \begin{array}{c} n \\ ((2^{i-3}-1)*n) \end{array} \\ n \quad ((2^{i-3}-1)*n) & \end{array}$$

Proof:

P_1 is immediately clear.

P_2 up to P_N we show by induction.

a) P_2 is found by stating the necessary conditions for the solution of P_1, according to Pontryagin's minimum principle. These conditions express the rational behaviour of player 1. These conditions are:

$$(5.3)\quad \begin{cases} x(t+1)-x(t)=Ax(t)+\sum_{j=1}^{N} B_j(u_j(t)-\hat{u}_j(t))+\sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t), \\ \qquad t=t_o,t_o+1,\dots,t_f-1, \\ x(t_o)=x_o, \end{cases}$$

$$(5.4)\quad \begin{cases} p_1(t)-p_1(t+1)=F'Q_1(Fx(t)+Gz_2(t)-\hat{y}_1(t))+A'p_1(t+1), \\ \qquad t=t_f-1,t_f-2,\dots,t_o, \\ p_1(t_f)=F'Q_1(Fx(t_f)+Gz_2(t_f)-\hat{y}_1(t_f)) \end{cases}$$

and

$$(5.5)\quad R_{11}(u_1(t)-\hat{u}_1(t))+B_1'\,p_1(t+1)=o, t=t_o,t_o+1,..,t_f-1.$$

Now we substitute (5.5) into (5.3). It is easily checked, that this leads to the required form for P_2.

b) P_{i+1} is found by stating the necessary conditions for the solution of P_i, according to Pontryagin's minimum principle. These conditions express the rational behaviour of player i.
First we make some preliminary steps.
Step 1: we write the constraints of P_i as a system with mixed boundary conditions by writing the backward recursive equations as forward recursive equations.

$$(5.6)\quad \tilde{p}_i(t+1)=(I+\tilde{A}_i')^{-1}(\tilde{p}_i(t)-\tilde{Q}_i\tilde{x}_i(t)-w_i(t)),t=t_o,t_o+1,..,t_f-1.$$

This leads to the system

$$(5.7)\quad \begin{bmatrix}\tilde{x}_i\\ \tilde{p}_i\end{bmatrix}(t+1)-\begin{bmatrix}\tilde{x}_i\\ \tilde{p}_i\end{bmatrix}(t)=\left(\begin{bmatrix}\tilde{A}_i & o\\ o & -I\end{bmatrix}+\begin{bmatrix}\tilde{S}_i\\ I\end{bmatrix}(I+\tilde{A}_i')^{-1}[-\tilde{Q}_i \vdots I]\right)\begin{bmatrix}\tilde{x}_i\\ \tilde{p}_i\end{bmatrix}(t)-$$

$$-\begin{bmatrix}\tilde{S}_i\\ I\end{bmatrix}(I+\tilde{A}_i')^{-1}w_i(t)+\begin{bmatrix}v_i(t)\\ o\end{bmatrix},t=t_o,t_o+1,...,t_f-1,$$

with initial boundary condition :

$$[I \vdots o]\begin{bmatrix}\tilde{x}_i\\ \tilde{p}_i\end{bmatrix}(t_o)=\begin{bmatrix}x_o\\ o\\ \vdots\\ o\end{bmatrix}$$

and final boundary condition :

$$[-\tilde{Q}_i \vdots I]\begin{bmatrix}\tilde{x}_i\\ \tilde{p}_i\end{bmatrix}(t_f)=-w_i(t_f).$$

Step 2: we rewrite the cost functional J_i, using (ii), (iii), (viii) and (ix).

$$(5.8)\quad J_i = \sum_{t=t_o}^{t_f} \tfrac{1}{2}\{\tilde{x}_i'(t)V_i\tilde{x}_i(t)+2\tilde{x}_i'(t)\begin{bmatrix} F'Q_i(Gz_2(t)-\hat{y}_i(t)) \\ o \\ \vdots \\ o \end{bmatrix} +$$

$$+(Gz_2(t)-\hat{y}_i(t))'Q_i(Gz_2(t)-\hat{y}_i(t))\} +$$

$$+\sum_{t=t_o}^{t_f-1} \tfrac{1}{2}\{\tilde{p}_i'(t+1)U_i\tilde{p}_i(t+1)+(u_i(t)-\hat{u}_i(t))'R_{ii}(u_i(t)-\hat{u}_i(t)) +$$

$$+\sum_{j=i+1}^{N}(u_j(t)-\hat{u}_j(t))'R_{ij}(u_j(t)-\hat{u}_j(t))\}.$$

Step 1 and step 2 only consisted of rewriting problem P_i. We proceed with stating the Hamiltonian functional for P_i and with stating the necessary conditions for the solution of P_i.
<u>Step 3</u>: we define the Hamiltonian functional H_i, using (5.6).

$$(5.9)\quad H_i(t,\tilde{x}_i,\tilde{p}_i,u_i,p_i,q_i,y_i)=\tfrac{1}{2}\tilde{x}_i'V_i\tilde{x}_i + \tilde{x}_i'\begin{bmatrix} F'Q_i(Gz_2(t)-\hat{y}_i(t)) \\ o \\ \vdots \\ o \end{bmatrix} +$$

$$+\tfrac{1}{2}[\tilde{p}_i-\tilde{Q}_i\tilde{x}_i-w_i(t)]'(I+\tilde{A}_i)^{-1}U_i(I+\tilde{A}_i')^{-1}[\tilde{p}_i-\tilde{Q}_i\tilde{x}_i-w_i(t)] +$$

$$+\tfrac{1}{2}[u_i-\hat{u}_i(t)]'R_{ii}[u_i-\hat{u}_i(t)] +$$

$$+[p_i' \vdots q_i' \vdots y_i']\{(\begin{bmatrix} \tilde{A}_i & o \\ o & -I \end{bmatrix} + \begin{bmatrix} \tilde{S}_i \\ I \end{bmatrix}(I+\tilde{A}_i')^{-1}[-\tilde{Q}_i \vdots I])\begin{bmatrix} \tilde{x}_i \\ \tilde{p}_i \end{bmatrix} -$$

$$-\begin{bmatrix} \tilde{S}_i \\ I \end{bmatrix}(I+\tilde{A}_i')^{-1}w_i(t) + \begin{bmatrix} B_i(u_i-\hat{u}_i(t)) \\ o \\ \vdots \\ o \end{bmatrix} + \begin{bmatrix} \sum_{j=i+1}^{N} B_j(u_j(t)-\hat{u}_j(t)) + \sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t) \\ o \\ \vdots \\ o \end{bmatrix}\}$$

Step 4: according to Pontryagin's minimum principle there exist

$[p_i' \vdots q_i' \vdots y_i'](t), t=t_o, t_o+1, \ldots t_f, \alpha_o$ and α_1,

such that the following equations are satisfied (see e.q. [10]):

1) the contraints of problem P_i.

2) $$\frac{\partial H_i}{\partial u_i}(t,\tilde{x}_i(t),\tilde{p}_i(t),u_i(t),p_i(t+1),q_i(t+1),y_i(t+1))=o \Rightarrow$$

$$\Rightarrow R_{ii}[u_i(t)-\hat{u}_i(t)]+B_i'p_i(t+1)=o \Rightarrow$$

$$\Rightarrow u_i(t)=-R_{ii}^{-1}B_i'p_i(t+1)+\hat{u}_i(t), t=t_o,t_o+1,\ldots,t_f-1 \qquad (5.10)$$

3) the backward recursive adjoint equations

$$\begin{bmatrix} p_i \\ q_i \\ y_i \end{bmatrix}(t)-\begin{bmatrix} p_i \\ q_i \\ y_i \end{bmatrix}(t+1)=\frac{\partial H_i}{\partial \begin{bmatrix} \tilde{x}_i \\ \tilde{p}_i \end{bmatrix}}(t,\tilde{x}_i(t),\tilde{p}_i(t),u_i(t),p_i(t+1),q_i(t+1),y_i(t+1)) =$$

$$=\left(\begin{bmatrix} \tilde{A}_i' & o \\ o & -I \end{bmatrix}+\begin{bmatrix} -\tilde{Q}_i' \\ I \end{bmatrix}(I+\tilde{A}_i)^{-1}[\tilde{S}_i' \vdots I]\right)\begin{bmatrix} p_i \\ q_i \\ y_i \end{bmatrix}(t+1)+$$

$$+\begin{bmatrix} -\tilde{Q}_i' \\ \\ I \end{bmatrix}(I+\tilde{A}_i)^{-1}U_i(I+\tilde{A}_i')^{-1}[\tilde{p}_i(t)-\tilde{Q}_i\tilde{x}_i(t)-w_i(t)] +$$

$$+\begin{bmatrix} I \\ o \end{bmatrix}\left(V_i\tilde{x}_i(t)+\begin{bmatrix} F'Q_i(Gz_2(t)-\hat{y}_i(t)) \\ o \\ \vdots \\ o \end{bmatrix}\right).$$

$$t=t_f-1,t_f-2,\ldots,t_o. \qquad (5.11)$$

4) the transversality conditions

$$\begin{bmatrix} p_i \\ q_i \\ y_i \end{bmatrix} (t_o) = \begin{bmatrix} I \\ \\ o \end{bmatrix} \alpha_o \qquad (5.12)$$

$$\begin{bmatrix} p_i \\ q_i \\ y_i \end{bmatrix} (t_f) = \begin{bmatrix} -Q_i' \\ \\ I \end{bmatrix} \alpha_1 + \begin{bmatrix} I \\ \\ 0 \end{bmatrix} \left(V_i \tilde{x}_i(t_f) + \begin{bmatrix} F' Q_i (Gz_2(t_f) - \hat{y}_i(t_f)) \\ o \\ \vdots \\ o \end{bmatrix}\right) \qquad (5.13)$$

<u>Step 5</u>: we rewrite the necessary conditions; using (5.6) we can split (5.11) into

$$(5.14) \quad y_i(t) = (I + \tilde{A}_i)^{-1} \left(\tilde{S}_i' \begin{bmatrix} p_i \\ q_i \end{bmatrix} (t+1) + y_i(t+1) + U_i \tilde{p}_i(t+1)\right),$$

$t = t_o, t_o+1, .., t_f-1,$

and

$$(5.15) \quad \begin{bmatrix} p_i \\ q_i \end{bmatrix} (t) - \begin{bmatrix} p_i \\ q_i \end{bmatrix} (t+1) = \tilde{A}_i' \begin{bmatrix} p_i \\ q_i \end{bmatrix} (t+1) - \tilde{Q}_i' \, y_i(t) + V_i \tilde{x}_i(t) +$$

$$+ \begin{bmatrix} F' Q_i (Gz_2(t) - \hat{y}_i(t)) \\ o \\ \vdots \\ o \end{bmatrix}, \quad t = t_f-1, t_f-2, .., t_o;$$

from (5.14) we find

$$(5.16) \quad y_i(t+1) - y_i(t) = \tilde{A}_i y_i(t) - U_i \tilde{p}_i(t+1) - \tilde{S}_i' \begin{bmatrix} p_i \\ q_i \end{bmatrix} (t+1),$$

$t = t_o, t_o+1, .., t_f-1;$

from (5.12) and (5.13) we find

$$(5.17)\quad y_i(t_o)=o$$

and

$$(5.18)\quad \begin{bmatrix} p_i \\ q_i \end{bmatrix}(t_f) = -\tilde{Q}_i' y_i(t_f) + V_i \tilde{x}_i(t_f) + \begin{bmatrix} F' Q_i (G z_{2_o}(t_f) - \hat{y}_i(t_f)) \\ o \\ \vdots \\ o \end{bmatrix}$$

Now we substitute (5.10) into $v_i(t)$ and regroup the forward recursive equations (5.1),(5.16) and (5.17) as well as the backward recursive equations (5.2),(5.15) and (5.18). It is easily checked, that this leads to the required form for P_{i+1}. Q.E.D.

<u>Theorem 2:</u>

The N-level Stackelberg solution for the linear quadratic difference game with an open loop information structure is given by

$$(5.19)\quad \begin{cases} u_1^*(t) = [\underbrace{-R_{11}^{-1}B_1'}_{n} \vdots \underbrace{o \ldots o}_{((2^{N-1}-1)*n)}](K(t+1)\tilde{x}_{N+1}(t+1)+g(t+1))+\hat{u}_1(t) \\ u_j^*(t) = [\underbrace{o \ldots o}_{(2^{j-2}*n)} \vdots \underbrace{-R_{jj}^{-1}B_j'}_{n} \vdots \underbrace{o \ldots o}_{((2^{N-1}-2^{j-2}-1)*n)}](K(t+1)\tilde{x}_{N+1}(t+1)+g(t+1))+\hat{u}_j(t), \end{cases}$$

$$j=2,3,..,N,\ t=t_o,t_o+1,..,t_f-1$$

where

$\tilde{x}_{N+1}(t)$, $t=t_o,t_o+1,..,t_f$, is the solution of the system

$$(5.20)\begin{cases}\tilde{x}_{N+1}(t+1)=(I-\tilde{S}_{N+1}K(t+1))^{-1}((I+\tilde{A}_{N+1})\tilde{x}_{N+1}(t)+\tilde{S}_{N+1}g(t+1)+v_{N+1}(t)),\\ \qquad\qquad t=t_o,t_o+1,..,t_f-1\,,\\ \tilde{x}_{N+1}(t_o)=[\,x_o'\,\vdots\,0\ldots0]'\end{cases}$$

and

$K(t),t=t_o+1,t_o+2,..,t_f$, is the solution of the backward recursive matrix Riccati equations

$$(5.21)\begin{cases}K(t)=\tilde{Q}_{N+1}+(I+\tilde{A}_{N+1}')K(t+1)(I-\tilde{S}_{N+1}K(t+1))^{-1}(I+\tilde{A}_{N+1}),\\ \qquad\qquad t=t_f-1\,,t_f-2,\ldots,t_o+1\\ K(t_f)=\tilde{Q}_{N+1}\end{cases}$$

and

$g(t),t=t_o+1,t_o+2,..,t_f$, is the solution of the backward recursive tracking equations

$$(5.22)\begin{cases}g(t)=w_{N+1}(t)+(I+\tilde{A}_{N+1}')g(t+1)+\\ \qquad +(I+\tilde{A}_{N+1}')K(t+1)(I-\tilde{S}_{N+1}K(t+1))^{-1}(\tilde{S}_{N+1}g(t+1)+v_{N+1}(t)),\\ \qquad\qquad t=t_f-1\,,t_f-2\,,\ldots,t_o+1,\\ g(t_f)=w_{N+1}(t_f)\end{cases}$$

and

$\tilde{x}_{N+1}(.),v_{N+1}(.),w_{N+1}(.),\tilde{A}_{N+1},\tilde{S}_{N+1}$ and $\tilde{Q}_{N+1}$ are defined in the same way as $\tilde{x}_i(.),v_i(.),w_i(.),\tilde{A}_i,\tilde{S}_i$ and $\tilde{Q}_i$ for $i=3,4,..N$ in theorem 1.

Proof:

From theorem 1 we know, that we can find the N-level Stackelberg solution for the linear quadratic difference game with an open loop information structure by solving P_N, the optimal control problem for the leader. We can do this by following the same reasoning as for P_i in part (b) of the proof of theorem 1. This leads to the open loop Stackelberg solution

$$(5.23)\begin{cases} u_1^{*}(t) = [\underbrace{-R_{11}^{-1}\, B_1'}_{n}\vdots\underbrace{o\ldots o}_{((2^{N-1}-1)*n)}]\, \tilde{p}_{N+1}(t+1)+\hat{u}_1(t), \\ u_j^{*}(t)=[\underbrace{o\ldots o}_{(2^{j-2}*n)}\vdots\underbrace{-R_{jj}^{-1}B_j'}_{n}\vdots\underbrace{o\ldots o}_{((2^{N-1}-2^{j-2}-1)*n)}]\, \tilde{p}_{N+1}(t+1)+\hat{u}_j(t), j=2,3,..,N, \quad t=t_o,t_o+1,\ldots,t_f-1 \end{cases}$$

where

$\tilde{p}_{N+1}(.)$ is defined in the same way as $\tilde{p}_i(.)$ for i=3,4,..,N in theorem 1

and

$\tilde{p}_{N+1}(t),t=t_o+1,t_o+2,..,t_f$, is part of the solution of the two point boundary value problem

$$(5.24)\quad \begin{cases} \tilde{x}_{N+1}(t+1)-\tilde{x}_{N+1}(t)=\tilde{A}_{N+1}\tilde{x}_{N+1}(t)+\tilde{S}_{N+1}\tilde{p}_{N+1}(t+1)+v_{N+1}(t), t=t_o t_o+1,..,t_f-1, \\ \tilde{x}_{N+1}(t_o)=[x_o'\vdots o\ldots o]' \end{cases}$$

$$\begin{cases} \tilde{p}_{N+1}(t)-\tilde{p}_{N+1}(t+1)=\tilde{A}_{N+1}'\tilde{p}_{N+1}(t+1)+\tilde{Q}_{N+1}\tilde{x}_{N+1}(t)+w_{N+1}(t), t=t_f-1,t_f-2,..,t_o, \\ \tilde{p}_{N+1}(t_f)=\tilde{Q}_{N+1}\tilde{x}_{n+1}(t_f)+w_{N+1}(t_f). \end{cases}$$

This two point boundary value problem can be solved in the usual way by postulating the linear relationship
$\tilde{p}_{N+1}(t)=K(t)\tilde{x}_{N+1}(t)+g(t),t=t_o,t_o+1,\ldots,t_f$.

The recursive equations (5.20),(5.21) and (5.22) follow immediately.
Q.E.D.

Remarks:

1) The costs can immediately be calculated from the cost functionals, taking into account that
$y(t)=[\underbrace{F}_{n}\vdots\underbrace{o..o}_{((2^{N-1}-1)*n)}]\,\tilde{x}_{N+1}(t)+Gz_2(t),t=t_o,t_o+1,\ldots,t_f.$

2) It is not clear directly under which conditions in terms of the data of the game the matrices $(I-\tilde{S}_{N+1}K(.))$ are nonsingular, although we can prove in an indirect way, that the open loop Stackelberg solution exists (and is unique) under the conditions, that $Q_i(.)$ and $R_{ij}(.)$ are positive semi-definite and $R_{ii}(.)$ are positive definite. In section 6 of this paper we show, that the feedback Stackelberg solution exists (and is unique) under those conditions. Furthermore we use the fact, that for a one stage problem the open loop solution and the feedback solution are identical. We can transform our problem into a one stage problem by defining (see e.g. [8])

$\underline{x}(1) \triangleq [x'(t_o+1)x'(t_o+2)....x'(t_f)]'$

$\underline{x}(o) \triangleq [x_o' \quad x_o' .... x_o']'$

$\underline{u}_i(o) \triangleq [u_i'(t_o)u_i'(t_o+1)...u_i'(t_f-1)]'$, $i=1,2,..,N$, etc....

The cost matrices for this one stage problem are

$\underline{Q}_i(o) \triangleq \text{diag}\ (Q_i(t_o),o,...,o)$

$\underline{Q}_i(1) \triangleq \text{diag}\ (Q_i(t_o+1),Q_i(t_o+2),...,Q_i(t_f))$

$\underline{R}_{ij}(o) \triangleq \text{diag}\ (R_{ij}(t_o),R_{ij}(t_o+1),...,R_{ij}(t_f-1))$

Finally we note, that $\underline{Q}_i(.)$ and $\underline{R}_{ij}(o)$ are positive semi-definite, if $Q_i(.)$ and $R_{ij}(.)$ are positive semi-definite, and that $\underline{R}_{ii}(o)$ is positive definite, if $R_{ii}(.)$ are positive definite.
The method described here is not attractive, if the planning period is long, because in that case the dimensions of the problem become very big.

Example:

We will elucidate theorem 1 and theorem 2 by means of a simple two stage two person nonzero sum linear quadratic difference game.
The open loop Stackelberg solution for this game was also calculated

in a straightforward way in [8].

Cost functionals: $J_1=2x^2(1)+2x^2(2)+u_1^2(o)+u_1^2(1)$

$$J_2=x^2(1)+x^2(2)+u_2^2(o)+u_2^2(1)$$

System: $\begin{cases} x(i+1)=x(i)+u_1(i)+u_2(i), i=0,1, \\ x(o)=x_o \end{cases}$

or

$A=o, B_1=1, B_2=1, C=o, F=1, G=o,$

$R_{ij}=o, i\neq j, \hat{y}_1=\hat{y}_2=\hat{u}_1=\hat{u}_2=o,$

$Q_1(2)=4, Q_1(1)=4, Q_1(o)=o,$

$Q_2(2)=2, Q_2(1)=2, Q_2(o)=o,$

$R_{11}(1)=2, R_{11}(o)=2, R_{22}(1)=2, R_{22}(o)=2.$

First we construct $\tilde{A}_3, \tilde{S}_3(.), \tilde{Q}_3(.), v_3$ and w_3:

$$\tilde{A}_3=\begin{bmatrix} o & o \\ o & o \end{bmatrix}; \; \tilde{S}_3(1)=\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ o & \frac{1}{2} \end{bmatrix}; \tilde{S}_3(o)=\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} \\ o & \frac{1}{2} \end{bmatrix}; \; v_3=o;$$

$$\tilde{Q}_3(2)=\begin{bmatrix} 4 & o \\ 2 & -4 \end{bmatrix}; \quad \tilde{Q}_3(1)=\begin{bmatrix} 4 & o \\ 2 & -4 \end{bmatrix}; \quad \tilde{Q}_3(o)=\begin{bmatrix} o & o \\ o & o \end{bmatrix} \; ; \quad w_3=o.$$

From (5.21) we find

$$K(2)=\begin{bmatrix} 4 & 0 \\ 2 & -4 \end{bmatrix}; \; K(1)=\begin{bmatrix} 26/5 & 4/5 \\ 11/5 & -26/5 \end{bmatrix}$$

From (5.22) we find $g(2)=o; g(1)=o$.

From (5.20) we find

$$\tilde{x}_3(o)=\begin{bmatrix} x_o \\ o \end{bmatrix}; \; \tilde{x}_3(1)=\begin{bmatrix} 36/145 & x_o \\ 11/145 & x_o \end{bmatrix} \; ; \; \tilde{x}_3(2)=\begin{bmatrix} 13/145 & x_o \\ 8/145 & x_o \end{bmatrix}$$

From (5.19) we find the open loop Stackelberg solution

$u_1(o) = -\frac{98}{145} x_o ; u_1(1) = -\frac{26}{145} x_o;$

$u_2(o) = -\frac{11}{145} x_o ; u_2(1) = \frac{3}{145} x_o.$

6. Feedback Stackelberg Solution.

The feedback Stackelberg solution is by definition the solution for the game, that is found by means of dynamic programming. As always, when we apply dynamic programming in linear quadratic frameworks, we have quadratic value functions for all players and we will operate on the backward recursive equations for those value functions.

Notation: for notational convenience we won't write the time dependency of the data-matrices.

The value function $V_i(.,x)$ for player i, i=1,2,..,N, satisfies the backward recursive equations

$$V_i(t,x) = \min_{u_i(t)} \{ \tfrac{1}{2}[Fx+Gz_2(t)-\hat{y}_i(t)]' Q_i [Fx+Gz_2(t)-\hat{y}_i(t)] + \sum_{j=1}^{N} \tfrac{1}{2}[u_j(t)-\hat{u}_j(t)]' R_{ij} [u_j(t)-\hat{u}_j(t)] + V_i(t+1,(I+A)x+\sum_{j=1}^{N} B_j[u_j(t)-\hat{u}_j(t)] + \sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t))\},$$

$$t = t_f-1, t_f-2, .., t_o. \qquad (6.1)$$

$$V_i(t_f,x) = \tfrac{1}{2}[Fx+Gz_2(t_f)-\hat{y}_i(t_f)]' Q_i [Fx+Gz_2(t_f)-\hat{y}_i(t_f)] \qquad (6.2)$$

The optimal behaviour of player i, i=1,2,...,N, at time t can be found by differentiating the right hand side of (6.1) with respect to $u_i(t)$ and setting this derivative equal to zero. In remark 2 at page 37 we show, that the Hessian matrix is positive definite (because we assumed the matrices Q_i and R_{ij} to be positive semi-definite and the matrices R_{ii} to be positive definite), so that the $u_i(.)$, $i=1,2,..,N$, calculated according to the described procedure, will indeed minimize the cost functionals. Remember, that the Stackelberg solution concept presupposes, that the decision of player i, $i=1,2,..,N-1$, depends on the decisions of the players higher in the hierarchy, that is the players $i+1$ up to N.

To solve for the optimal behaviour of player i, $i=1,2..,N$, we define the quadratic value functions as follows:

$V_i(t,x)=\frac{1}{2}x'K_i(t)x+g_i'(t)x+c_i(t)$, $t=t_o,t_o+1,..,t_f$, $i=1,2,..,N$.

Without loss of generality we assume the K- matrices to be symmetric. The described procedure, that is the equalizing of the gradients with respect to $u_i(t)$ of the right hand sides of (6.1) with zero, leads to the following set of equations.

$$R_{11}[u_1(t)-\hat{u}_1(t)]+B_1'(K_1(t+1)((I+A)x+\sum_{j=1}^{N}B_j[u_j(t)-\hat{u}_j(t)]+$$
$$+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t))+g_1(t+1))=o \qquad (6.3)$$

and

$$R_{ii}[u_i(t)-\hat{u}_i(t)]+\sum_{j=1}^{i-1}\left(\frac{\partial u_j(t)}{\partial u_i(t)}\right)R_{ij}[u_j(t)-\hat{u}_j(t)]+(B_i'+\sum_{j=1}^{i-1}\frac{\partial u_j(t)}{\partial u_i(t)}B_j').$$
$$.(K_i(t+1)((I+A)x+\sum_{j=1}^{N}B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t))+g_i(t+1))=o,$$

$$i=2,3,..N \qquad (6.4),$$

$$t=t_o,t_o+1..,t_f-1.$$

Notation: the sum-term Σ and the product-term Π should be understood as follows:
if the index is nondecreasing, as normal, and if the index is decreasing, $\Sigma \equiv o$ and $\Pi \equiv I$.

Theorem 3:
The set of equations (6.3) and (6.4) have the following solution:

$$u_i(t)=-D_i(t)((I+A)x+\sum_{j=i+1}^{N} B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(-D_i^{(\ell)}(t))g_\ell(t+1)+\hat{u}_i(t),\, i=1,2,..,N,\, t=t_o,t_o+1,..,t_f-1 \qquad (6.5)$$

where

$$D_1^{(1)}(t)=(R_{11}+B_1'K_1(t+1)B_1)^{-1}B_1';\, D_1^{(\ell)}(t)=o,\, \ell=2,3,..,N \qquad (6.6)$$

$$D_1(t)=D_1^{(1)}(t)K_1(t+1) \qquad (6.7)$$

$t=t_o,t_o+1,..,t_f-1$

and

$$D_i(t)=M_i^{-1}(t)B_i'\{\sum_{j=1}^{i-1}(\prod_{k=j+1}^{i-1}(I-B_kD_k(t)))'D_j'(t)R_{ij}\cdot$$

$$\cdot D_j(t)(\prod_{k=j+1}^{i-1}(I-B_kD_k(t)))+$$

$$+(\prod_{k=1}^{i-1}(I-B_kD_k(t)))'K_i(t+1)(\prod_{k=1}^{i-1}(I-B_kD_k(t)))\} \qquad (6.8)$$

$$D_i^{(\ell)}(t)=M_i^{-1}(t)B_i'\{\sum_{j=1}^{i-1}(\prod_{k=j+1}^{i-1}(I-B_kD_k(t)))'D_j'(t)R_{ij}\cdot$$

$$\cdot(D_j^{(\ell)}(t)-D_j(t)\sum_{m=j+1}^{i-1}(\prod_{k=j+1}^{m-1}(I-B_kD_k(t)))B_mD_m^{(\ell)}(t))+$$

$$+(\prod_{k=1}^{i-1}(I-B_kD_k(t)))'K_i(t+1)\cdot$$

$$\cdot(\sum_{m=1}^{i-1}(\prod_{k=1}^{m-1}(I-B_kD_k(t)))(-B_mD_m^{(\ell)}(t)))\},$$

$\ell=1,2,...,i-1$ (6.9)

$$D_i^{(i)}(t)=M_i^{-1}(t)B_i'(\prod_{k=1}^{i-1}(I-B_kD_k(t)))' \qquad (6.10)$$

$$D_i^{(\ell)}(t)=o, \ell=i+1,i+2,..,N \qquad (6.11)$$

where

$$M_i(t)=R_{ii}+B_i'\{\sum_{j=1}^{i-1}(\prod_{k=j+1}^{i-1}(I-B_kD_k(t)))'D_j'(t)R_{ij}.$$

$$.D_j(t)(\prod_{k=j+1}^{i-1}(I-B_kD_k(t)))+$$

$$+(\prod_{k=1}^{i-1}(I-B_kD_k(t)))'K_i(t+1)(\prod_{k=1}^{i-1}(I-B_kD_k(t)))\}B_i \qquad (6.12)$$

for $i=2,3,..,N$, $t=t_o,t_o+1,...,t_f-1$

Before we prove theorem 3 we will first state and prove some useful lemmas.

Lemma 1:

a) $I-\sum_{i=j}^{n}P_i(\prod_{k=i+1}^{n}(I-P_k))=\prod_{k=j}^{n}(I-P_k)$

b) $\sum_{i=j}^{n}\{\tilde{P}_i-P_i\sum_{k=i+1}^{n}(\prod_{\ell=i+1}^{k-1}(I-P_\ell))\tilde{P}_k\}=\sum_{k=j}^{n}(\prod_{\ell=j}^{k-1}(I-P_\ell))\tilde{P}_k$

Proof:

a) $I-\sum_{i=j}^{n}P_i(\prod_{k=i+1}^{n}(I-P_k))=(I-P_n)-P_{n-1}(I-P_n)-$

$$-P_{n-2}(I-P_{n-1})(I-P_n)-...-P_j(\prod_{k=j+1}^{n}(I-P_k))=$$

$$=\prod_{k=j}^{n}(I-P_k), \text{Q.E.D.}$$

b) $$\sum_{i=j}^{n}\tilde{P}_i - \sum_{i=j}^{n} P_i \sum_{k=i+1}^{n}\left(\prod_{\ell=i+1}^{k-1}(I-P_\ell)\right)\tilde{P}_k =$$

$$= \sum_{i=j}^{n}\tilde{P}_i - \sum_{k=j+1}^{n}\sum_{i=j}^{k-1} P_i\left(\prod_{\ell=i+1}^{k-1}(I-P_\ell)\right)\tilde{P}_k =$$

$$= \tilde{P}_j + \sum_{k=j+1}^{n}\left(I - \sum_{i=j}^{k-1} P_i\left(\prod_{\ell=i+1}^{k-1}(I-P_\ell)\right)\right)\tilde{P}_k =$$

$$\underset{\substack{\uparrow\\ \text{lemma 1a.}}}{=} \tilde{P}_j + \sum_{k=j+1}^{n}\left(\prod_{\ell=j}^{k-1}(I-P_\ell)\right)\tilde{P}_k =$$

$$= \sum_{k=j}^{n}\left(\prod_{\ell=j}^{k-1}(I-P_\ell)\right)\tilde{P}_k, \text{ Q.E.D.}$$

Lemma 2:

For $j=2,3,..,N, t=t_o,t_o+1,..,t_f-1$ we have:
suppose, that (6.5) is correct for $i=1,2,..,j-1$, then

$$\frac{\partial u_i(t)}{\partial u_j(t)} = -B_j'\left(\prod_{k=i+1}^{j-1}(I-B_kD_k(t))\right)'D_i'(t), i=1,2,..,j-1 \qquad (6.13)$$

Proof:

We prove this by backward induction:

1) $i=j-1$: $$\frac{\partial u_{j-1}(t)}{\partial u_j(t)} \underset{\substack{\uparrow\\ (6.5)}}{=} -B_j'D_{j-1}'(t).$$

2) Suppose, that (6.13) is correct for $i=j-1,j-2,..,\ell+1$;

$$i=\ell: \frac{\partial u_\ell(t)}{\partial u_j(t)} \underset{\substack{\uparrow\\ (6.5)}}{=} -\left(\sum_{i=\ell+1}^{j-1}\left(\frac{\partial u_i(t)}{\partial u_j(t)}\right)L_i' + B_j'\right) D_\ell'(t)=$$

$$= -\Big(\sum_{i=\ell+1}^{j-1} \big(-B_j'\big(\prod_{k=i+1}^{j-1}(I-B_kD_k(t))\big)'D_i'(t)\big)B_i'+B_j'\Big)D_\ell'(t)=$$

↑ induction assumption

$$= -B_j'\Big(I-\sum_{i=\ell+1}^{j-1}\big(\prod_{k=i+1}^{j-1}(I-B_kD_k(t))\big)'D_i'(t)B_i'\Big)D_\ell'(t)=$$

$$= -B_j'\Big(\prod_{k=\ell+1}^{j-1}(I-B_kD_k(t))\Big)'D_\ell'(t).$$

↑ lemma 1a

Q.E.D.

<u>Lemma 3:</u>

For $j=2,3,..,N+1, t=t_o,t_o+1,..,t_f-1$ we have:

suppose, that (6.5) is correct for $i=1,2,..,j-1$, then

$$u_i(t)-\hat{u}_i(t)=-D_i(t)\Big(\prod_{k=i+1}^{j-1}(I-B_kD_k(t))\Big).$$

$$.\Big((I+A)x+\sum_{m=j}^{N}B_m[u_m(t)-\hat{u}_m(t)]+\sum_{m=1}^{N}B_m\hat{u}_m(t)+Cz_1(t)\Big)+$$

$$+\sum_{\ell=1}^{N}\Big(-D_i^{(\ell)}(t)+D_i(t)\sum_{m=i+1}^{j-1}\Big(\prod_{k=i+1}^{m-1}(I-B_kD_k(t))\Big)B_mD_m^{(\ell)}(t)\Big).$$

$$.\ g_\ell(t+1),\ i=1,2,..,j-1. \qquad (6.14)$$

<u>Proof:</u>

We prove this by backward induction.

1) $i=j-1$: (6.5) immediately implies (6.14).

2) Suppose, that (6.14) is correct for $i=j-1,j-2,..,n+1$;

$i=n$:

$$u_n(t)-\hat{u}_n(t) \underset{(6.5)}{=} -D_n(t)((I+A)x+\sum_{m=n+1}^{N} B_m[u_m(t)-\hat{u}_m(t)]+\sum_{m=1}^{N} B_m\hat{u}_m(t)+$$

$$+Cz_1(t))+\sum_{\ell=1}^{N}(-D_n^{(\ell)}(t))g_\ell(t+1)=$$

$$\underset{\substack{\text{induction}\\\text{assumption}}}{=} -D_n(t)((I+A)x+\sum_{m=n+1}^{j-1} B_m\{-D_m(t)(\prod_{k=m+1}^{j-1}(I-B_kD_k(t))).$$

$$.((I+A)x+\sum_{\ell=j}^{N} B_\ell[u_\ell(t)-\hat{u}_\ell(t)]+\sum_{\ell=1}^{N} B_\ell\hat{u}_\ell(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(-D_m^{(\ell)}(t)+D_m(t)\sum_{i=m+1}^{j-1}(\prod_{k=m+1}^{i-1}(I-B_kD_k(t)))B_iD_i^{(\ell)}(t))g_\ell(t+1)\}+$$

$$+\sum_{m=j}^{N} B_m[u_m(t)-\hat{u}_m(t)]+\sum_{m=1}^{N} B_m\hat{u}_m(t)+Cz_1(t))+\sum_{\ell=1}(-D_n^{(\ell)}(t))g_\ell(t+1)=$$

$$=-D_n(t)\{I-\sum_{m=n+1}^{j-1} B_mD_m(t)(\prod_{k=m+1}^{j-1}(I-B_kD_k(t)))\}.$$

$$.((I+A)x+\sum_{\ell=j}^{N} B_\ell[u_\ell(t)-\hat{u}_\ell(t)]+\sum_{\ell=1}^{N} B_\ell\hat{u}_\ell(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}\{-D_n^{(\ell)}(t)+D_n(t)(\sum_{m=n+1}^{j-1}(B_mD_m^{(\ell)}(t)-$$

$$-B_mD_m(t)\sum_{i=m+1}^{j-1}(\prod_{k=m+1}^{i-1}(I-B_kD_k(t)))B_iD_i^{(\ell)}(t)))\}g_\ell(t+1).$$

From this, lemma 1a and lemma 1b, (6.14) for i=n is immediately clear.

Q.E.D.

Now we are ready to prove theorem 3.

Proof theorem 3:

We prove this by induction.

1) We rewrite (6.3) as follows:

$$u_1(t)=-(R_{11}+B_1'K_1(t+1)B_1)^{-1}B_1'(K_1(t+1)((I+A)x+$$

$$+\sum_{j=2}^{N} B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N} B_j\hat{u}_j(t)+Cz_1(t))+g_1(t+1))+\hat{u}_1(t),$$

$$t=t_o,t_o+1,..t_f-1.$$

From this, (6.5) for i=1, (6.6) and (6.7) are immediately clear.

2) Suppose, that the solution of (6.3) and (6.4) for i=1,2,..,j-1 is given by (6.5) up to (6.12) for i=1,2,..,j-1, then, using lemma 2 and lemma 3, we can rewrite (6.4) for i=j as follows:

$$R_{jj}[u_j(t)-\hat{u}_j(t)]+\sum_{i=1}^{j-1}\{-B_j'(\prod_{k=i+1}^{j-1}(I-B_kD_k(t)))'D_i'(t)\}R_{ji}.$$

$$.\{-D_i(t)(\prod_{k=i+1}^{j-1}(I-B_kD_k(t)))((I+A)x+\sum_{m=j}^{N}B_m[u_m(t)-\hat{u}_m(t)]+$$

$$+\sum_{m=1}^{N}B_m\hat{u}_m(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(-D_i^{(\ell)}(t)+D_i(t)\sum_{m=i+1}^{j-1}(\prod_{k=i+1}^{m-1}(I-B_kD_k(t)))B_mD_m^{(\ell)}(t))g_\ell(t+1)\}+$$

$$+(B_j'+\sum_{i=1}^{j-1}\{-B_j'(\prod_{k=i+1}^{j-1}(I-B_kD_k(t)))'D_i'(t)\}B_i').$$

$$.(K_j(t+1)((I+A)x+\sum_{i=1}^{j-1}B_i\{-D_i(t)(\prod_{k=i+1}^{j-1}(I-B_kD_k(t))).$$

$$.((I+A)x+\sum_{m=j}^{N} B_m[u_m(t)-\hat{u}_m(t)]+\sum_{m=1}^{N} B_m\hat{u}_m(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(-D_i^{(\ell)}(t)+D_i(t)\sum_{m=i+1}^{j-1}(\prod_{k=i+1}^{m-1}(I-B_kD_k(t)))B_mD_m^{(\ell)}(t))g_\ell(t+1)\}+$$

$$+\sum_{i=j}^{N} B_i[u_i(t)-\hat{u}_i(t)]+\sum_{i=1}^{N} B_i\hat{u}_i(t)+Cz_1(t))+g_j(t+1))=o,$$

$t=t_o,t_o+1,..,t_f-1.$

From lemma 1a we know, that

$$B_j'(I-\sum_{i=1}^{j-1}(\prod_{k=i+1}^{j-1}(I-B_kD_k(t)))'D_i'(t)B_i')=$$

$$=B_j'(\prod_{k=1}^{j-1}(I-B_kD_k(t)))',t=t_o,t_o+1,..,t_f-1.$$

From lemma 1b we know, that

$$\sum_{i=1}^{j-1}\{-B_iD_i^{(\ell)}(t)+B_iD_i(t)\sum_{m=i+1}^{j-1}(\prod_{k=i+1}^{m-1}(I-B_kD_k(t)))B_mD_m^{(\ell)}(t)\}=$$

$$=\sum_{m=1}^{j-1}(\prod_{k=1}^{m-1}(I-B_kD_k(t)))(-B_mD_m^{(\ell)}(t)),t=t_o,t_o+1,..,t_f-1.$$

Now it is easy to derive (6.5) and (6.8) up to (6.12) for i=j.

Q.E.D.

In fact we have found in theorem 3 the feedback Stackelberg solution in terms of $K_i(.)$ and $g_i(.)$,i=1,2,..,N.
In theorem 4 we will state and prove backward recursive equations for $K_i(.)$ and $g_i(.)$,i=1,2,..,N.

Theorem 4:

The feedback Stackelberg solution is given by (6.5) up to (6.12), if $K_i(.)$, $i=1,2,..,N$, satisfies the backward recursive matrix Riccati equations

$$K_i(t)=F'Q_iF+(I+A)'\{\sum_{j=1}^{N}(\prod_{k=j+1}^{N}(I-B_kD_k(t)))'D_j'(t)R_{ij}\cdot$$

$$\cdot D_j(t)(\prod_{k=j+1}^{N}(I-B_kD_k(t)))+(\prod_{k=1}^{N}(I-B_kD_k(t)))'K_i(t+1)\cdot$$

$$\cdot(\prod_{k=1}^{N}(I-B_kD_k(t)))\}(I+A),t=t_f-1,t_f-2,\dots,t_o+1 \qquad (6.15)$$

$$K_i(t_f)=F'Q_iF \qquad (6.16)$$

and $g_i(.)$, $i=1,2,..,N$, satisfies the backward recursive tracking equations

$$g_i(t)=F'Q_i(Gz_2(t)-\hat{y}_i(t))+(I+A')\cdot$$

$$\cdot\{\sum_{j=1}^{N}(\prod_{k=j+1}^{N}(I-B_kD_k(t)))'D_j'(t)R_{ij}(D_j(t)(\prod_{k=j+1}^{N}(I-B_kD_k(t)))\cdot$$

$$\cdot(\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(D_j^{(\ell)}(t)-D_j(t)\sum_{m=j+1}^{N}(\prod_{k=j+1}^{m-1}(I-B_kD_k(t)))B_mD_m^{(\ell)}(t))g_\ell(t+1))+$$

$$+(\prod_{k=1}^{N}(I-B_kD_k(t)))'(K_i(t+1)((\prod_{k=1}^{N}(I-B_kD_k(t)))(\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}\sum_{m=1}^{N}\Big(\prod_{k=1}^{m-1}(I-B_kD_k(t))\Big)(-B_mD_m^{(\ell)}(t))g_\ell(t+1))+g_i(t+1)),$$

$$t=t_f-1,t_f-2,\ldots,t_o+1 \tag{6.17}$$

$$g_i(t_f)=F'Q_i(Gz_2(t_f)-\hat{y}_i(t_f)). \tag{6.18}$$

Proof:

Remember, that we had the quadratic value functions

$V_i(t,x)=\frac{1}{2}x'K_i(t)x+g_i'(t)x+c_i(t), t=t_o,t_o+1,\ldots,t_f, i=1,2,\ldots,N.$
Now (6.2) immediately implies (6.16) and (6.18).

From (6.1), theorem 3 and lemma 3 for $j=N+1$ we know

$$\frac{1}{2}x'K_i(t)x+g_i'(t)x+c_i(t)=\frac{1}{2}\sum_{j=1}^{N}[u_j(t)-\hat{u}_j(t)]'R_{ij}[u_j(t)-\hat{u}_j(t)]+$$

$$+\frac{1}{2}[Fx+Gz_2(t)-\hat{y}_i(t)]'Q_i[Fx+Gz_2(t)-\hat{y}_i(t)]+$$

$$+\frac{1}{2}[(I+A)x+\sum_{j=1}^{N}B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t)]'\cdot$$

$$\cdot K_i(t+1)[(I+A)x+\sum_{j=1}^{N}B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t)]+$$

$$+g_i'(t+1)[(I+A)x+\sum_{j=1}^{N}B_j[u_j(t)-\hat{u}_j(t)]+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t)]+$$

$$+c_i(t+1), t=t_f-1,t_f-2,\ldots,t_o+1,$$

where

$$u_j(t)-\hat{u}_j(t)=-D_j(t)\Big(\prod_{k=j+1}^{N}(I-B_kD_k(t))\Big)((I+A)x+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t))+$$

$$+\sum_{\ell=1}^{N}(-D_j^{(\ell)}(t)+D_j(t))\sum_{m=j+1}^{N}\left(\prod_{k=j+1}^{m-1}(I-B_kD_k(t))\right)B_mD_m^{(\ell)}(t))g_\ell(t+1).$$

Comparing quadratic terms in x and linear terms in x leads to (6.15) and (6.17), respectively, using lemma 1a and lemma 1b in the same way as in the proof of theorem 3.
Q.E.D.

Remarks:

1) If one is interested in the costs of the game, one should also evaluate the constant terms $c_i(.),i=1,2,..,N$.
2) By inductive reasoning we can show, that the Hessian matrices $(R_{11}+B_1'K_1(.)B_1)$ in (6.6) and $M_i(.),i=2,3,..,N$, in (6.12) are positive definite, hence nonsingular, so that minimum costs, $D_i(.)$ and $K_i(.),i=1,2,..,N$, exist.
   First we note from (6.16), that $K_i(t_f)$ are positive semi-definite, because Q_i are positive semi-definite.
   Next we will describe the induction steps in t and i .
   Suppose $K_i(t+1)$ are positive semi-definite. Because R_{ij} are positive semi-definite and R_{ii} are positive definite, we can conclude, that
a) $(R_{11}+B_1'K_1(t+1)B_1)$ is positive definite, so that from (6.6) and (6.7) $D_1(t)$ exists,
b) if $D_i(t)$, $i=1,2,..,j-1$, exist, then from (6.12) $M_j(t)$ is positive definite, so that from (6.8) $D_j(t)$ exists.
   From (6.15) we note, that, if $D_i(t)$, $i=1,2,..N$, exist, then $K_i(t)$ are positive semi-definite, because Q_i, R_{ij} and $K_i(t+1)$ are positive semi-definite. Q.E.D.
3) The system, driven by the feedback Stackelberg solution, can be written as follows, using lemma 3 for j=N+1 and lemma 1a:

$$\begin{cases} x(t+1)=\left(\prod_{k=1}^{N}(I-B_kD_k(t))\right)\left((I+A)x(t)+\sum_{j=1}^{N}B_j\hat{u}_j(t)+Cz_1(t)\right), t=t_o,t_o+1,..t_f-1 \\ \\ x(t_o)=x_o \end{cases}$$

4) The algorithm for the feedback Stackelberg solution has a loop backward in time, consisting of the equations (6.15) up to (6.18) and (6.6) up to (6.12), and a loop forward in time, consisting of (6.5) and the system equations.

<u>Example:</u>

We will elucidate theorem 3 and theorem 4 by means of the same example as was used on page 24.

First we note from (6.17) and (6.18), that $g_1(2)=g_2(2)=g_1(1)=g_2(1)=0$.

So, it is of no use to calculate $D_i^{(\ell)}(t), i=1,2, \ell=1,2, t=0,1$.

From (6.16) we find

$$K_1(2)=4; K_2(2)=2.$$

From (6.6)and (6.7) we find $D_1(1) = \frac{2}{3}$.

From (6.12) we find $M_2(1)=\frac{20}{9}$.

From (6.8) we find $D_2(1)=\frac{1}{10}$.

From (6.15) we find

$$K_1(1)=\frac{127}{25}; K_2(1)=\frac{11}{5}.$$

From (6.6) and (6.7) we find $D_1(o) = \frac{127}{177}$.

From (6.12) we find $M_2(o)=\frac{68158}{31329}$.

From (6.8) we find $D_2(o)=\frac{2750}{34079}$.

From (6.5) and the system equations we find the feedback Stackelberg solution

$$u_2(o)= - \frac{2750}{34079}x_o; u_1(o)= - \frac{22479}{34079}x_o;$$

$u_2(1) = -\ \frac{885}{34079}x_o; u_1(1) = -\ \frac{5310}{34079}x_o,$

with trajectory $x_o, \frac{8850}{34079}x_o, \frac{2655}{34079}x_o.$

8. Conclusion.

In this paper solutions are derived for the N-level Stackelberg linear quadratic difference game with exogenous inputs, nonfeasible ideal paths and a fixed time horizon for two information structures. A topic of further research could be the closed loop information structures for this game. In a second paper on this subject we will consider more players on the same decision level and we will give an application in the form of a small linked econometric model for the Common Market.

Acknowledgement.

I would like to thank Joseph Plasmans from Tilburg and Frank Claeys from Gent for reading the manuscript.

References.

1. Plasmans, J.E.J., "Linked Econometric Models as a Difference Game; Nash Optimality", Research Memorandum, Tilburg University, 1978.
2. Plasmans, Joseph E.J. and de Zeeuw, Aart J., "Pareto Optimality and Incentives to Cooperate in Linear Quadratic Difference Games", Research Memorandum FEW 75, Tilburg University, September 1978.
3. Medanic, J. and Radojevic, D., "Multilevel Stackelberg Strategies in Linear-Quadratic Systems", JOTA, vol. 24, pp. 485-497, 1978.
4. Gardner, jr., B.F. and Cruz, jr.,J.B., "Feedback Stackelberg Strategy for M-level Hierarchical Games", IEEE, vol. AC-23, no. 3, pp. 489-491, June 1978.
5. Cruz, jr., Jose B., "Leader-Follower Strategies for Multilevel Systems", IEEE, vol. AC-23, no.2, pp. 244-254, April 1978.
6. Simaan M. and Cruz, jr., J.B., "On the Stackelberg Strategy in Nonzero-

Sum Games", JOTA, vol. 11, no. 5, pp. 533-555, 1973.

7. Simaan M. and Cruz, jr., J.B., "Additional Aspects of the Stackelberg Strategy in Nonzero-Sum Games", JOTA, vol.11, no. 6, pp. 613-626, 1973.
8. Olsder, Geert Jan, "Information Structures in Differential Games", in "Differential Games and Control Theory II", edited by Emilio O. Roxin, Pan-Tai Liu and Robert L. Sternberg, Marcel Dekker, Inc., New York and Basel, pp. 99-135, 1977.
9. Medanic, J., "Closed-Loop Stackelberg Strategies in Linear-Quadratic Problems", IEEE, vol. AC-23, no. 4, pp. 632-637, August 1978.
10. Varaiya, P.P., "Notes on Optimization", Van Nostrand Reinhold Company, New York, 1972.